Generating ANTAB $T_{\rm sys}$ Files: `antabfs`

Bob Campbell, JIVE

1. Antenna calibration review

- A source of flux density S delivers a power to the output terminals of the receiver of

$$P = \frac{1}{2}\,A_{\rm eff}\,S\,\Delta\nu \tag{1}$$

$$= \frac{1}{2}\,\eta_A\,A_{\rm geom}\,S\,\Delta\nu \tag{2}$$

  where η_A is the *aperture efficiency* (the ratio between effective area and geometric area for the antenna at that frequency/pointing direction).

- A resistive load at a temperature T delivers a power of

$$P = k\,T\,\Delta\nu \tag{3}$$

  where k is the Boltzmann constant ($1.308{\times}10^{-23}$ J/K).

- The *antenna temperature* T_A is the equivalent temperature of a resistive load providing the same power at the output terminals of the receiver as the source of flux density S:

$$T_A = \frac{1}{2k}\,\eta_A\,A_{\rm geom}\,S$$

$$= \frac{\pi D^2}{8k}\,\eta_A\,S \tag{4}$$

  where the last line expresses $A_{\rm geom} = \pi D^2/4$.

2. System temperature & SEFD

- The *system temperature* $T_{\rm sys}$ is the equivalent temperature of a resistive load providing the same power at the output terminals of the receiver as the system noise.
- The *system equivalent flux density* SEFD is the flux density of a fictitious source providing the same poser at the output terminals of the receiver as the system noise.

- These two quantities can be related to each other via a *sensitivity* Γ: $T_{\rm sys} = \Gamma\ SEFD$, with Γ in units of Kelvins per Jansky. From equation (4),

$$\Gamma = \frac{\pi D^2}{8k}\,\eta_A \tag{5}$$

3. Y-method for measuring system temperature

- One way to measure the $T_{\rm sys}$ of a receiver is to put two different-temperature resistive loads "into" the antenna and measure the ratio of the output powers:

$$\begin{aligned} P_{\rm hot} &= (T_{\rm hot} + T_{\rm sys})\,g \\ P_{\rm cold} &= (T_{\rm cold} + T_{\rm sys})\,g \end{aligned} \tag{6}$$

  where g is the gain of the receiver — here we take it that the receiver remains in a linear regime for both loads, such that g is independent of input/output power.

- Forming the ratio of these two output powers: $Y = P_{\rm hot}/P_{\rm cold}$ (Y should be > 1) and solving for $T_{\rm sys}$ yields:

$$T_{\rm sys} \quad = \quad \frac{T_{\rm hot} - Y T_{\rm cold}}{Y - 1} \tag{7}$$

4. Field-system power measurements

- The *cal signal* is a broad-band noise source at a specific temperature (`caltemp`). Ideally, this is constant, but is really a function of frequency and time — calibrating these changes is why the `rxg` files are important.
- Total-power integrators (Mark4) include:
  - `tpi`: measured when cal-source off
  - `tpical`: measured when cal-source on
  - `tpzero`: zero levels (must be subtracted from `tpi` or `tpical` before using them in calculations)
- The cal source "fires" only when there is a long-enough gap in the schedule (*i.e.,* when the disk isn't recording).
  - `tpi`$'$: a `tpi` measurement close(st) in time to the cal-source firing.

- The difference (`tpical` – `tpi′`) will be an important quantity, essentially setting the scale between TPI counts and physical temperature. Sometimes this is called `tpdiff`.

↪ How do you get $T_{\rm sys}$ from these FS-log quantities?

- Take the output with the cal-source off and on to be:

$$\begin{aligned} (T_{\rm sys})\, g &= \texttt{tpi} - \texttt{tpzero} \\ (T_{\rm sys} + T_{\rm cal})\, g &= \texttt{tpical} - \texttt{tpzero} \end{aligned} \tag{8}$$

- Forming the ratio of these two powers and solving for $T_{\rm sys}$, and expressing the result in terms of the TPI counts leads to:

$$T_{\rm sys} = T_{\rm cal}\, \frac{\texttt{tpi} - \texttt{tpzero}}{\texttt{tpical} - \texttt{tpi}'} \tag{9}$$

- Representative `tpical`–`tpi′` value ~ 1000.
  - too low ↦ larger scatter; more editing in `antabfs` may be needed
  - ~ 0 ↦ dead cal source(?)
  - jumps ↦ change in attenuation; unstable cal source

- VLBA4 racks are slightly different, with an automatic gain control (AGC) active, keeping the power-counts in a good range for the samplers. The reported `tpgain` is an index for a look-up table for AGC gain level (256 entries). The values from this table can be used in a formula to compute a `tpi` proxy given a `tpi′`:

$$\texttt{tpi} = \texttt{tpi}' \cdot e^{0.23026 \cdot (GL' - GL)} \tag{10}$$

  where GL and GL' are the gain-levels from the look-up table indexed by the logged `tpgain` and `tpgain′` values.

- For the EVN, $T_{\rm cal}$ of the cal-sources is measured in CL experiments that take place in each frequency-band sub-session during disk-based EVN sessions. For EVN stations, $T_{\rm cal}$ has clear frequency dependence, even within a band, and can vary from session to session. The `rxg` files result from such measurements; see the *Antenna Gain Calibration* workshop (M. Lindqvist) for details.

5. What's the astronomer want?

- We now have $T_{\mathrm{sys}}(t)$ within an experiment
  - `tpical` – `tpi`′ provides an physical tie to the temperature of the cal-source at gaps in the schedule
  - `tpi` provides a relative scale in between gaps

- SEFD as a function of time will give the astronomer the noise in flux-density units (Jy) of his telescopes:

$$SEFD(t) = \frac{T_{\mathrm{sys}}(t)}{\mathrm{GAIN}} = \frac{T_{\mathrm{sys}}(t)}{DPFU \times POLY(elev)} \tag{11}$$

  where
  - $DPFU$ = degrees per flux unit [K/Jy] — another measure of antenna sensitivity.
  - $POLY$ = the *gain curve*, above as a function elevation (could also be zenith angle [90° – elevation] or some more complicated dependencies for equatorial-mount stations). The largest contributor to gain changing with elevation is gravitation distortion of the antenna.

- The astronomer will eventually need to convert the dimensionless correlation coefficients output from the correlator into physical flux-density units, which will depend on the geometric means of the SEFDs of the two stations forming each baseline. Thus inaccuracies in any of $T_{\mathrm{sys}}(t)$, DPFU, or the gain curve could translate into inaccuracies in the amplitudes of the visibility data. Since the map of the source will ultimately result from the Fourier transform of the visibility data (via various stages of image reconstruction procedures), such inaccuracies will result in errors/spurious features in the source structure in the "map plane".

6. antabfs: where to get it

- `antabfs.pl` is a perl program that inputs a FS log and `rxg` files to:
  - compute/edit `tpical` – `tpi`′ values for each VC/BBC
  - compute/edit resulting T_{sys}
  - output an *antabfs* file

  It needs the `pgplot` perl module for its on-screen editing and plotting functions. The path to this should be set in the environment variable `PERL5LIB`.

- It was written originally by Cormac Reynolds, and subsequently supported/updated by Giuseppe Cimò and Jun Yang — all at JIVE. We are always interested in getting feedback for new features or other modifications that would improve its usefulness to the stations.

- You can get it via anonymous ftp:

  ftp.jive.nl

  cd outgoing/yang

  get antabfs.tar.gz

- The current tarball expands to a subdirectory `antabfs2010` underneath your current working directory. There is a guide in this at `antabfs2010/docs/antabfs.tex`.

7. RXG files

- more authoritative information in *Antenna Gain Calibration* workshop — M. Lindqvist

- $T_{\rm cal}$ can be function of frequency and time (see fig.1).

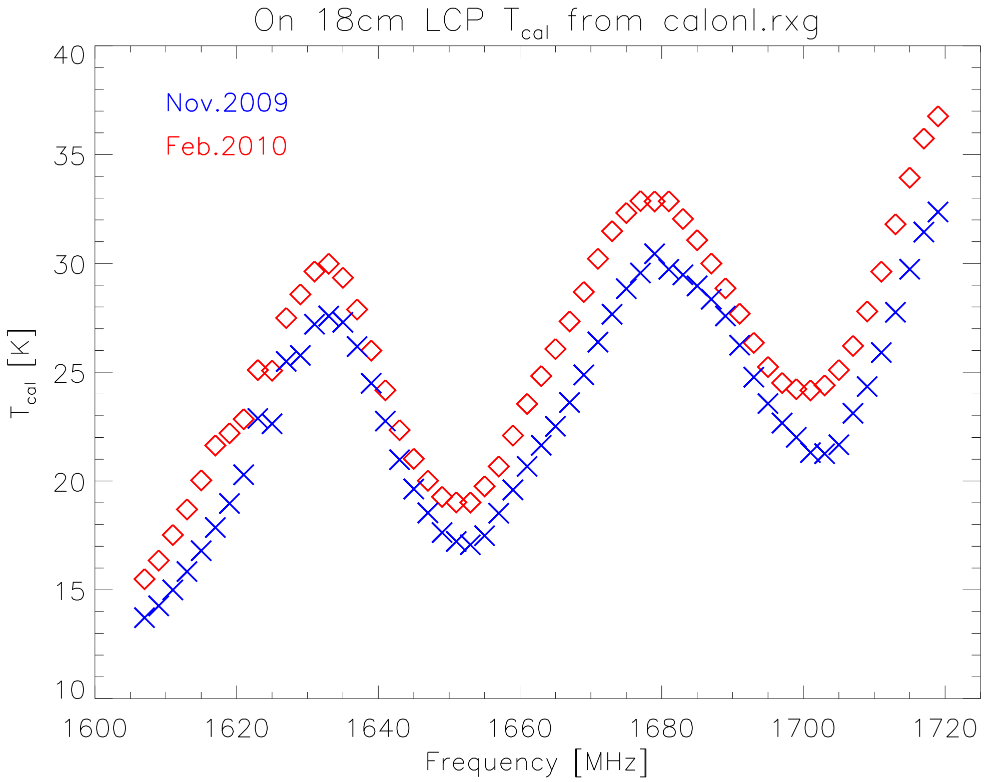

Figure 1: $T_{\rm cal}$ for On 18 cm LCP.

- Abbreviated Contents:

ℓ.1 : applicable LO-frequency range

ℓ.2 : file creation date

ℓ.3 : beam width (in terms of frequency or physical degrees)

ℓ.4 : available polarizations (one of or both: rcp, lcp)

ℓ.5 : DPFU [K/Jy] for each polarization listed in ℓ.4

$\ell.6$: gain curve

argument = elevation or zenith angle via `TYPE`

max # coefficients = 10

$\ell.7$: rows of: POL FREQ $T_{\rm cal}$

max # rows = 100

group rows by polarization, then sort by increasing frequency

terminate with "`end_tcal_table`"

$\ell.8$: receiver temperature (if 0, no opacity corrections used)

usually 0 below K-band

$\ell.9$: spill-over table: rows of ELEV [deg] T of spill-over noise

terminate with "`end_spillover_table`"

8. Calibration feed-back for experiments correlated at JIVE

- Each experiment correlated at JIVE gets pipelined. The *a priori* amplitude calibration for this comes directly from the stations' *antabfs* output files.
- Pipeline results reside in the EVN archive, which you can access at

  `www.jive.nl/select-experiment`

  by clicking on the experiment you want from the pull-down menu. Go to the Pipeline tab, then to the pipeline plots link (top one of the table). Find the section beginning "Amplitude corrections". This will provide a list of sources in the experiment, each with a `pdf`, a text file, and a statistical summary.

  - The `pdf` file plots amplitude correction factors (1 = no correction needed) by station/subband/polarization — see fig.2.
  - The statistical summary tabulates by station/channel the median amplitude correction factor and related statistics. It also provides the antenna numbering that is needed to investigate the text file.
  - The text file contains the raw information from which the statistical summary was computed. It tabulates per subband/polarization the amplitude correction factor ($\times$10000), with time-sampling roughly on a per-scan basis. The telescopes form the columns, labeled by the antenna number in the experiment.

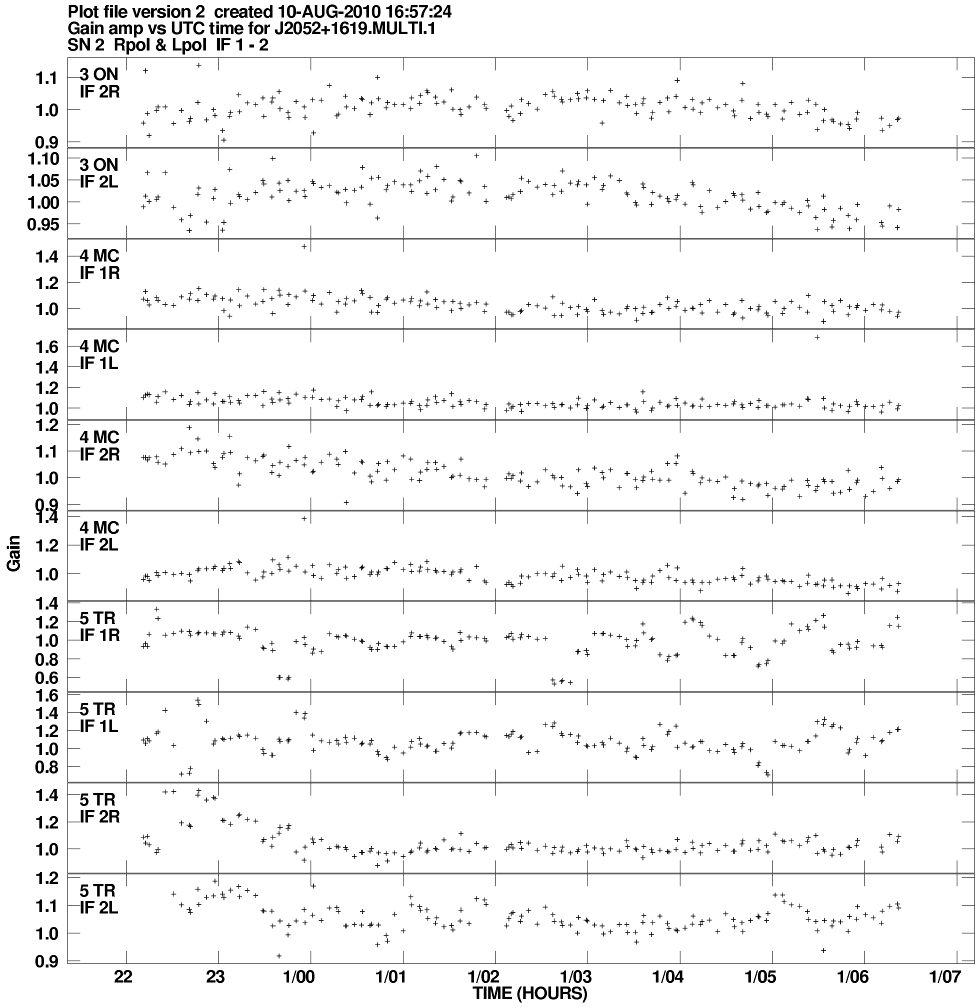

Figure 2: page from a amplitude correction plot from the EVN archive (experiment ES064B, 18 cm)

9. antabfs demo

- You may need to make a couple quick adjustments to the `antabfs.pl` program before running:
  - update the first line for the location of your perl
  - update the `$gaindir` definition to reflect where your *rxg* files will be found.

- The basic syntax for `antabfs.pl` is just:

  `antabfs.pl` *logfile*

  A variety of command line switches are available; inclusion of the switch `-h` will produce a full listing of these.

- Once the FS-log has been read, you will be prompted for your plotting device for the (possibly) interactive plots. Responding with a "?" returns a list of available devices. `/xs` is the usual response (short for /xserve — a persistent xwindow).

[1] In the first part of the program, it will loop through all VCs, showing `tpdiff` = `tpical`−`tpi`′ as a function of time, together with a red piece-wise linear fit (hereafter, PWLF) and lines representing ±5 deviations from the fit (green & blue). You have some control over how the fit is made.

- Hitting return while a plot is showing deletes the `tpdiff` values outside the green & blue range. A new PWLF will be generated and plotted.
  - You can control interactively how many deviations these lines span by simply entering a different integer and hitting return. A new PWLF will be generated, but the blue & green lines will revert to ±5 deviations.
  - You can control interactively how many `tpdiff` points go into each PWLF bin by hitting "B" and responding to the query.
- If you want more control over deleting `tpdiff` points, you can type "E" and hit return. You will then be able to draw boxes on the plot via:
  - left mouse button (or "A") — selects the two opposite corners of a rectangle
  - middle mouse button (or "D") — restarts the current rectangle
  - "C" — cancels all previously defined rectangles
  - "Z"/"U" — makes a zoom box / unzooms
  - right mouse button (or "X") — exits, deleting all the points in all the defined rectangles
- When you're finally happy with this VC's plot, enter "Y" to accept the changes made and move on to the next VC (entering "Y" as the first step bypasses any editing on the displayed VC).

[2] Once you've edited the `tpdiff` values for all the VCs, the program will take some time to compute $T_{\rm sys}$, and then loop through the VCs again displaying $T_{\rm sys}(t)$.

- The only editing option you have here is the "E" option described above — interactively defining rectangles inside of which points will be deleted.

[3] Once you've gone through the $T_{\rm sys}$ for all the VCs, the program will make a color-coded plot of all the VCs' $T_{\rm sys}$ plotted together, to allow a check for global consistency.

- It will prompt you whether you want to go back to step [2] to re-edit the individual-VC $T_{\rm sys}$ values. If you don't want to do this, then the program will finish by making the *antabfs* file.

- There are a couple associated plotting programs. `plot_tsys.pl` takes an *antabfs* file and generates a plot similar to that made in step [3], with the option of putting each VC onto separate panels.